**Model Parameters :**

**Definition: Parameters are the internal variables of the model that are learned from the training data during the training process. They are the "weights" and "biases" in a model that get adjusted to minimize the loss function and improve model predictions.**

**Hyperparameters :**

**Definition: Hyperparameters are external configurations of the model that are set before the learning process begins. They control the training process and the architecture of the model but are not learned from the data.**

**Hyperparameter Tuning ?**

* Hyperparameter tuning is the process of selecting the optimal values for a [machine learning](https://www.geeksforgeeks.org/machine-learning/) model’s hyperparameters.
* Hyperparameters control the learning process of the model, such as the learning rate, the number of neurons in a neural network, or the kernel size in a support vector machine.
* The goal of hyperparameter tuning is to find the values that lead to the best performance on a given task.

**Hyperparameters in Neural Networks**

* **Learning rate:** Too small a learning rate can result in slow convergence, while too large a learning rate can lead to instability and divergence.
* **Epochs:** This hyperparameter represents the number of times the entire training dataset is passed through the model during training. Increasing the number of epochs can improve the model’s performance.
* **Number of Layers**: A neural network is made up of vertically arranged components, which are called layers. There are mainly input layers, hidden layers, and output layers. A 3-layered neural network gives a better performance than a 2-layered network.
* **Activation function:** This hyperparameter introduces non-linearity into the model, allowing it to learn complex decision boundaries. Common activation functions include sigmoid, tanh, and Rectified Linear Unit (ReLU).

**Types of Hyperparameter:**

**Manual Hyperparameter Tuning:**

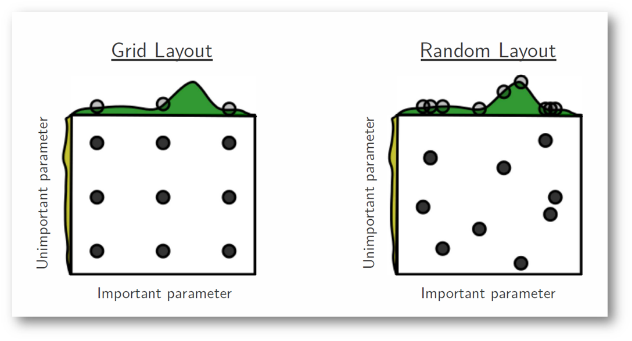
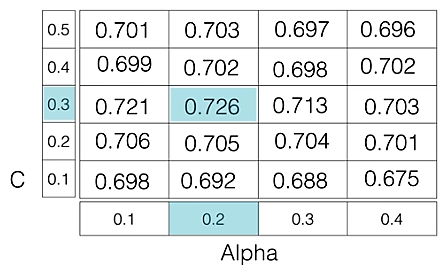
**Definition**: Manual hyperparameter tuning involves adjusting hyperparameter values based on intuition, domain expertise, and trial-and-error. It is a time-consuming process and requires multiple training and evaluation iterations to find the best settings.

**Techniques:**

**Grid Search**  is a method used to find the best hyperparameters for a model by trying out all possible combinations of the given options. It builds the model for each combination, checks how well it performs, and picks the combination that gives the best results.

For example, if we are tuning two parameters, C = [0.1, 0.2, 0.3] and Alpha = [0.1, 0.2], GridSearchCV will test all combinations like C = 0.1, Alpha = 0.1, C = 0.2, Alpha = 0.2, etc., and choose the one with the best performance.

It's accurate but can be slow because it tests every possible combination.



**Output:**

Tuned Logistic Regression Parameters: {'C': 0.006105402296585327}

Best score is 0.853

**Randomized Search** is a method for hyperparameter tuning that randomly selects combinations of hyperparameters to test, instead of exhaustively trying all possibilities like Grid Search. In each iteration, it picks a random set of values, evaluates the model's performance, and returns the best combination after a fixed number of trials.

This approach is faster and more efficient than Grid Search because it skips unnecessary computations and focuses on random sampling. Despite being random, it often produces results similar to Grid Search but in less time.

**Example**:  
For learning\_rate = [0.01, 0.05, 0.1] and batch\_size = [16, 32, 64], random search may randomly pick combinations like:

* learning\_rate =0.05,batch\_size=32
* learning\_rate =0.01,batch\_size=64, skipping many combinations.

**Automatic Hyperparameter Tuning :**

**Definition**: Automatic hyperparameter tuning uses algorithms or tools to systematically search for the best hyperparameter values without manual intervention. Techniques like grid search, random search, Bayesian optimization, and genetic algorithms are commonly used.

**Techniques:**

**Bayesian Optimization** is an advanced method for hyperparameter tuning that uses past results to guide the search for the best hyperparameters. Instead of testing combinations randomly or exhaustively, it models the problem as an optimization task and uses a probabilistic function to predict which hyperparameters are likely to perform best.

**Example**:  
If the model finds that a learning rate of 0.050.050.05 works better than 0.010.010.01 in early evaluations, Bayesian optimization might prioritize testing values near 0.050.050.05 (e.g., 0.045,0.0550.045, 0.0550.045,0.055) instead of randomly testing other ranges.

**Challenges in Hyperparameter Tuning**

* Dealing with High-Dimensional Hyperparameter Spaces: Efficient Exploration and Optimization
* Handling Expensive Function Evaluations: Balancing Computational Efficiency and Accuracy
* Incorporating Domain Knowledge: Utilizing Prior Information for Informed Tuning
* Developing Adaptive Hyperparameter Tuning Methods: Adjusting Parameters During Training

**Applications of Hyperparameter Tuning**

* Model Selection: Choosing the Right Model Architecture for the Task
* Regularization Parameter Tuning: Controlling Model Complexity for Optimal Performance
* Feature Preprocessing Optimization: Enhancing Data Quality and Model Performance
* Algorithmic Parameter Tuning: Adjusting Algorithm-Specific Parameters for Optimal Results

**Advantages of Hyperparameter tuning:**

* Improved model performance
* Reduced overfitting and underfitting
* Enhanced model generalizability
* Optimized resource utilization
* Improved model interpretability

**Disadvantages of Hyperparameter tuning:**

* Computational cost
* Time-consuming process
* Risk of overfitting
* Requires expertise